Efficient Similarity Queries via Lossy Compression

Idoia Ochoa, Amir Ingber and Tsachy Weissman

Dept. of Electrical Engineering, Stanford University, Stanford CA 94305
Email: {iochoa, ingber, tsachy}@stanford.edu

***Abstract*—The generation of new databases and the amount of data on existing ones is growing exponentially. As a result, executing queries on large databases is becoming a timely and challenging task. With this in mind, we study the problem of compressing sequences in a database so that similarity queries can still be performed efficiently on the compressed database. The fundamental limits of this problem characterize the tradeoff between compression rate and the reliability of the queries performed on the compressed data. While those asymptotic limits have been studied and characterized in past work, how to approach these limits in practice has remained largely unexplored. In this paper, we propose an approach to this task, based in part on existing lossy compression algorithms.**

Specifically, we consider queries of the form: *"which sequences in the database are similar to a given sequence* $\mathbf{y}$*?"*. For the case where similarity between sequences is measured via Hamming distortion, we construct schemes whose performance is close to the fundamental limits. Furthermore, we test our scheme on a sample database of real DNA sequences, and show significant compression while still allowing highly reliable query answers.

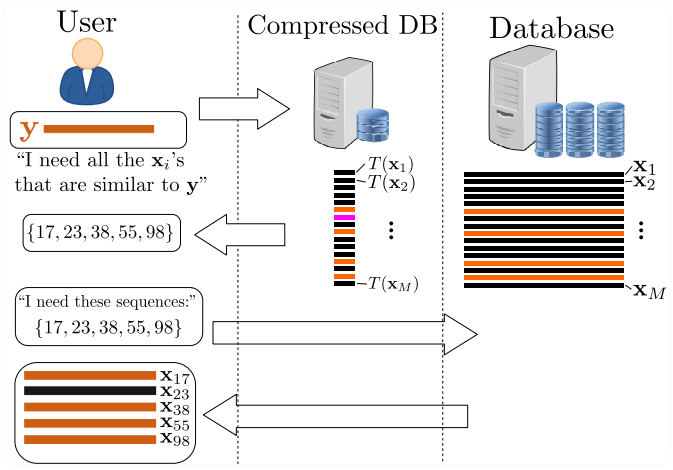

Fig. 1. Answering queries from signatures: a user first makes a query to the compressed database, and upon receiving the indexes of the sequences that may possibly be similar, discards the false positives by retrieving the sequences from the original database.

I. Introduction

The amount of data that we store is growing exponentially rapidly. As a result, large databases are being generated. Databases consisting of genomic data are one important example. The biological database *Genbank* contains almost 200 million DNA sequences [1], and the database BIOZON contains well over 100 million records [2]. Due to the size of the databases, performing queries on them is a challenging task.

With this challenge in mind, in this paper we study the problem of compressing a database so that queries about the original data can be answered efficiently given only the compressed version. Such compression enables the database to support a large amount of search queries: due to the smaller size of the database, it will become possible to store the compressed database in several locations, thus facilitating the access to the database. Further, the smaller size may reduce the time needed to execute a query.

Given a database consisting of many sequences, we focus on queries of the form: *"which sequences in the database are similar to a given sequence* $\mathbf{y}$*?"*. This kind of query is of practical interest in many applications, such as molecular phylogenetics, where relationships among species are established by the similarity between their respective DNA sequences. More specifically, for each sequence $\mathbf{x}$ in the database, our scheme will only keep a short *signature* of fixed-length, denoted by $T(\mathbf{x})$. Queries are then performed using the given query sequence $\mathbf{y}$ and the signature $T(\mathbf{x})$.

When answering a query from the signature alone, one cannot hope for an exact answer every time. There are two types of errors that can be made: a *false positive*, when the compressed database identifies a sequence as similar to the query sequence when it is actually not similar, and a *false negative*, when the answer to the query is that the sequences are not similar, but they actually are. We impose the restriction that false negatives are not permitted, since such errors cannot be detected. On the other hand, false positives do not cause an error *per se*, as they only introduce a computational burden due to the need of further verification. Fig. 1 shows a typical usage case.

This problem was studied from an information-theoretic perspective in [3] for discrete sources, and in [4], [5] for Gaussian sources. In this setting, both the query sequence and the database sequence are assumed to be drawn randomly and independently from a given distribution. Those papers show that there is a fundamental tradeoff between compression rate, sequence length, and reliability of queries performed on compressed data. However, in those papers the achievability proofs are non-constructive, which raises the question of how to design such schemes.

In this paper we propose a framework for compression for similarity queries, based on lossy compression algorithms. While our scheme relies on (possibly suboptimal) compression algorithms, we have the guarantee for zero false negatives,

as required. With our scheme, the signature of a sequence $\mathbf{x}$ consists of a compressed description of a reconstruction sequence $\hat{\mathbf{x}}$ (the output of the lossy compressor), and additional information that enables to answer the question "are $\mathbf{x}$ and $\mathbf{y}$ similar?" with zero false negatives. This side information can be, for example, the actual distortion between $\mathbf{x}$ and $\hat{\mathbf{x}}$, $d(\mathbf{x}, \hat{\mathbf{x}})$. We consider different types of such side information and also efficient ways to represent it.

For the case of a binary symmetric source and Hamming distortion, the fundamental limits are computable and provide the basis for performance evaluation. We evaluate our scheme on randomly generated i.i.d. data and show results that are comparable to those limits. We then test our scheme on a sample database of DNA sequences (taken from the BIOZON [2] database), and show significant compression while at the same time allowing reliable query answers. For example, when the sample database is compressed by more than a factor of 5 (over 80% compression ratio), the fraction of the original database that has to be retrieved is less than 0.1%.

The rest of the paper is organized as follows. In Section II we formalize the problem and extend some of the theoretical results presented in the past to our setting. The proposed architecture is introduced in Section III, and the simulation results for binary sources are shown in Section IV. Section V considers the extension to q-ary alphabets and shows the result for the DNA database. Finally, we give concluding remarks in section VI.

II. Problem Formulation

A. Notation and Problem Description

We begin by introducing the notation. Let upper case, lower case, and calligraphic letters denote, respectively, random variables, values they may assume, and their alphabets. In this paper we deal only with discrete sources. Therefore, without loss of generality, we assume $\mathcal{X} = \{0, 1, \ldots, |\mathcal{X}| - 1\}$, where $|\mathcal{X}|$ denotes the cardinality of the alphabet. We use boldface to denote vectors. $\mathbf{x}$ is a shorthand for a vector of length n, and $[1 : k]$ denotes the set $\{1, 2, \ldots, k\}$.

Given two sequences $\mathbf{x}$ and $\mathbf{y}$, we measure their *similarity* by applying a distortion function $d : (\mathbf{x}, \mathbf{y}) \to \mathcal{R}^+$ that operates symbol by symbol. We say that two sequences $\mathbf{x}$ and $\mathbf{y}$ are *D-similar* when $d(\mathbf{x}, \mathbf{y}) \leq D$. For Hamming distortion,

$$d(\mathbf{x}, \mathbf{y}) \triangleq \frac{1}{n} d(x_i, y_i) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{x_i \neq y_i\}}, \tag{1}$$

where $\mathbb{1}_{\{A\}}$ is the indicator function that takes value one if A is true and zero otherwise.

The goal is to design an architecture to compress each sequence in the database such that similarity queries can still be performed efficiently on the compressed database. Specifically, given a query sequence $\mathbf{y}$, we seek those sequences $\mathbf{x}$ in the database that are *D-similar* to $\mathbf{y}$. We want our design to be admissible (i.e., without false negatives) and reliable (with a small amount of false positives).

B. Formal Definitions

A rate-R identification system (T, g) consists of a signature assignment $T : \mathcal{X}^n \to [1 : 2^{nR}]$, and a query function $g : [1 : 2^{nR}] \times \mathcal{X}^n \to \{\texttt{no}, \texttt{maybe}\}$. The notational choice of `no` and `maybe` reflects the fact that false negatives are not allowed, while false positives are possible. In other words, if the answer to the query is `no` one can conclude that the corresponding sequences $\mathbf{x}$ and $\mathbf{y}$ are not *D-similar*, while nothing can be implied about the similarity of the sequences if the answer is `maybe`. A system is said to be *D-admissible* (i.e., without false negatives) if

$$g(T(\mathbf{x}), \mathbf{y}) = \texttt{maybe}, \ \forall \mathbf{x}, \mathbf{y} \text{ s.t. } d(\mathbf{x}, \mathbf{y}) \leq D. \tag{2}$$

With these definitions in mind, for a database $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)}\}$, the compressed database $\tilde{\mathcal{D}}$ is given by $\tilde{\mathcal{D}} = \{T(\mathbf{x}^{(1)}), \ldots, T(\mathbf{x}^{(M)})\}$, and a query Q returns

$$Q(\mathbf{y}, \tilde{\mathcal{D}}) = \{i : g(T(\mathbf{x}^{(i)}), \mathbf{y}) = \texttt{maybe}, 1 \leq i \leq M\}.$$

C. Theoretical Framework

Let $\mathbf{X}$ and $\mathbf{Y}$ be random vectors, representing the sequence from the database and the query sequence, respectively. Unless specified otherwise, we assume that $\mathbf{X}$ and $\mathbf{Y}$ are independent, with entries drawn independently from the same distribution P_X. For a given similarity threshold D, there is a tradeoff between compression rate and reliability. Next we give some definitions that we borrow from [4], [5].

Definition 1: For a source distribution P_X and similarity threshold D, a rate R is said to be *D-achievable* if there exists a sequence of rate-R admissible schemes $(T^{(n)}, g^{(n)})$, s.t.

$$\lim_{n \to \infty} \Pr\left\{ g^{(n)}\left(T^{(n)}(\mathbf{X}), \mathbf{Y}\right) = \texttt{maybe} \right\} = 0. \tag{3}$$

Definition 2: For a similarity threshold D, the *identification rate* $R_{\mathrm{ID}}(D)$ is the infimum of D-achievable rates. That is,

$$R_{\mathrm{ID}}(D) \triangleq \inf\{R : R \text{ is } D\text{-achievable}\}. \tag{4}$$

Having defined $R_{\mathrm{ID}}(D)$, the rate at which the probability of `maybe` can be made to vanish is also of significant interest. We expect this rate to be exponential as in the traditional source coding setting, motivating the following definition:

Definition 3: Fix $R \geq R_{\mathrm{ID}}(D)$. The *identification exponent* $\mathbf{E}_{\mathrm{ID}}(R)$ is defined as

$$\limsup_{n \to \infty} -\frac{1}{n} \log \Pr\left\{ g^{(n)}\left(T^{(n)}(\mathbf{X}), \mathbf{Y}\right) = \texttt{maybe} \right\},$$

where $g^{(n)}, T^{(n)}$ represent the optimal schemes at rate R and length n. Note that the quantity $\mathbf{E}_{\mathrm{ID}}(R)$ implicitly depends on the parameters of the problem (i.e., D and P_X).

D. Fundamental Limits

Characterizing the identification rate and exponent is a hard problem in general. In [3], where the variable-length coding equivalent of our setting was considered, the authors present an achievable rate (they do not consider the converse to the identification rate problem). The results of [3] for the identification exponent rely on an auxiliary random

variable of unbounded cardinality, thus making the quantities uncomputable in general. For the quadratic-Gaussian case the identification rate and exponent were found in [4] [5]. One of the few other cases in which the identification rate and exponent can be found is the binary symmetric source with Hamming distortion:

Theorem 1: For a binary symmetric source ($P_X(x=0) = P_X(x=1) = 0.5$) and Hamming distortion, the identification rate is given by

$$R_{\mathrm{ID}}(D) = \begin{cases} 1 - H(0.5 - D) & \text{for } D < 0.5 \\ 1 & \text{for } D \geq 0.5, \end{cases} \tag{5}$$

where $H(p) = -p\log(p) - (1-p)\log(1-p)$ is the binary entropy function.

Theorem 2: For Hamming distortion and a symmetric binary source,

$$\mathbf{E}_{\mathrm{ID}}(R) = \begin{cases} 0, & 0 \leq R \leq R_{\mathrm{ID}}(D); \\ 1 - H(D + H^{-1}(1-R)), & R_{\mathrm{ID}}(D) \leq R \leq 1; \\ 1 - H(D), & R \geq 1. \end{cases} \tag{6}$$

Proof of Theorems 1 and 2: Appendix A ∎

III. Proposed architecture

In this section we introduce the main contribution of this paper, an architecture that is *D-admissible* and that can work for any discrete database. Specifically, we show how lossy compression algorithms can be used to compute $T(\mathbf{x})$, and decision rules to decide whether a sequence may or may not satisfy $d(\mathbf{x},\mathbf{y}) \leq D$. These rules ensure that the system is *D-admissible*, regardless of any probabilistic assumptions.

A. Compression Scheme

The proposed compression scheme is based on fixed-length lossy compression algorithms (see [6], [7] and references therein). They are characterized by an encoding function $f_n : \mathbf{x} \to [1 : 2^{nR'}]$ and a decoding function $g_n : [1 : 2^{nR'}] \to \hat{\mathbf{x}}$, where $\hat{\mathbf{x}} = g_n(f_n(\mathbf{x}))$ denotes the reconstructed sequence.

The proposed architecture stores, together with the output $i \in [1 : 2^{nR'}]$ of the lossy compression algorithm, extra information denoted as side-information. Specifically, we can store the distortion between $\hat{\mathbf{x}}$ and $\mathbf{x}$. Therefore, we can think of the signature as $T : \mathbf{x} \to \{i, d(\mathbf{x},\hat{\mathbf{x}})\}$. The total rate of the system is $R = R' + \Delta R$, where ΔR represents the extra bits to store the distortion. The general scheme is depicted in Fig 2.

A variant of this scheme consists of storing the *joint type* $P_{\mathbf{x},\hat{\mathbf{x}}}$ instead of the distortion. Specifically, for the binary case

$$P_{\mathbf{x},\hat{\mathbf{x}}} = \frac{1}{n}[N(00|\mathbf{x},\hat{\mathbf{x}}), N(01|\mathbf{x},\hat{\mathbf{x}}), N(10|\mathbf{x},\hat{\mathbf{x}}), N(11|\mathbf{x},\hat{\mathbf{x}})],$$

where $N(ab|\mathbf{x},\hat{\mathbf{x}})$ (hereafter N_{ab}) denotes the number of occurrences of (a,b) in $(\mathbf{x},\hat{\mathbf{x}})$. Notice that in this case the signature provides more information, since the distortion can be computed from the type. The only drawback is that representing the joint type requires more bits. We discuss the representation of the side information in detail in section III-C.

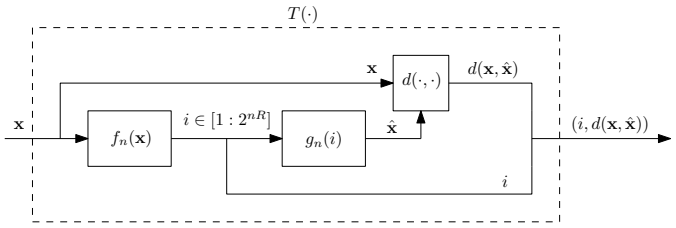

Fig. 2. Proposed scheme to compute $T(\mathbf{x})$, with the side information given by the distortion $d(\mathbf{x},\hat{\mathbf{x}})$.

B. Decision Rules

Recall that a general decision function $g : [1 : 2^{nR}] \times \mathcal{X}^n \to \{\mathtt{no}, \mathtt{maybe}\}$ must satisfy (2). As described before, we consider signatures of the form $(i, d(\mathbf{x},\hat{\mathbf{x}}))$ and $(i, P_{\mathbf{x},\hat{\mathbf{x}}})$, from which $\hat{\mathbf{x}}$ can be recovered. We now introduce the decision rules for each case.

Case 1: $T(\mathbf{x}) = (i, d(\mathbf{x},\hat{\mathbf{x}}))$.

For ease of notation, in this section we denote $d(\mathbf{x},\hat{\mathbf{x}})$ by d. The decision function is given by

$$g((i,d),\mathbf{y}) = \begin{cases} \mathtt{maybe}, & d - D \leq d(\hat{\mathbf{x}},\mathbf{y}) \leq d + D; \\ \mathtt{no}, & \text{otherwise.} \end{cases} \tag{7}$$

For any given distortion satisfying the triangle inequality, the above satisfies (2). For binary symmetric sources and Hamming distortion, the probability of `maybe` for each sequence is readily verified to be analytically given by

$$\Pr\{\mathtt{maybe}|T(\mathbf{x})\} = \sum_{i=\lceil n(d-D)\rceil}^{\lfloor n(d+D)\rfloor} \binom{n}{i} 2^{-n}. \tag{8}$$

Interestingly, (8) does not depend on $\hat{\mathbf{x}}$ (this is unique to the symmetric binary-Hamming case).

Case 2: $T(\mathbf{x}) = (i, P_{\mathbf{x},\hat{\mathbf{x}}})$.

For any $\hat{\mathbf{x}} \in \hat{\mathcal{X}}^n$, define the set $A(\hat{\mathbf{x}})$ as the set of all $\mathbf{x}$'s with a given joint type Q:

$$A_Q(\hat{\mathbf{x}}) \triangleq \{\mathbf{x} \in \mathcal{X}^n : (\mathbf{x},\hat{\mathbf{x}}) \text{ have the joint type } Q\}.$$

The decision rule, based on $(i, P_{\mathbf{x},\hat{\mathbf{x}}})$, is

$$g(T(\mathbf{x}),\mathbf{y}) = \begin{cases} \mathtt{maybe}, & \exists \mathbf{x}' \in A_{P_{\mathbf{x},\hat{\mathbf{x}}}}(\hat{\mathbf{x}}) : d(\mathbf{x}',\mathbf{y}) \leq D; \\ \mathtt{no}, & \text{otherwise,} \end{cases} \tag{9}$$

where on the right hand side $\hat{\mathbf{x}}$ is the reconstruction associated with index i. The above decision rule satisfies (2), but is somewhat abstract for implementation purposes. We simplify it for the case of binary sources and Hamming distortion. Given $d_{\min}(\hat{\mathbf{x}}, P_{\mathbf{x},\hat{\mathbf{x}}}, \mathbf{y}) \triangleq \min_{\mathbf{x}' \in A_{P_{\mathbf{x},\hat{\mathbf{x}}}}(\hat{\mathbf{x}})} d(\mathbf{x}',\mathbf{y})$, (9) can be rewritten as

$$g((i, P_{\mathbf{x},\hat{\mathbf{x}}}),\mathbf{y}) = \begin{cases} \mathtt{maybe}, & d_{\min}(\hat{\mathbf{x}}, P_{\mathbf{x},\hat{\mathbf{x}}}, \mathbf{y}) \leq D; \\ \mathtt{no}, & \text{otherwise.} \end{cases}$$

Given $\hat{\mathbf{x}}$, define I_0 and I_1 as follows: $I_0 \triangleq \{i : \hat{x}_i = 0\}$ and $I_1 \triangleq \{i : \hat{x}_i = 1\}$. Given $\mathbf{y}$, define $j_0 = |\{i \in I_0 : y_i = 1\}|$ and $j_1 = |\{i \in I_1 : y_i = 1\}|$. With this notation, it is easy to verify that $d_{\min}(\hat{\mathbf{x}}, P_{\mathbf{x},\hat{\mathbf{x}}}, \mathbf{y})$ can be written as

$$d_{\min}(\hat{\mathbf{x}}, P_{\mathbf{x},\hat{\mathbf{x}}}, \mathbf{y}) = \frac{1}{n}\left[|N_{10} - j_0| + |N_{11} - j_1|\right],$$

where N_{10} and N_{11} are known from the type. Note that the entire type can be computed from $\{N_{10}, N_{11}\}$, since the sequence $\hat{\mathbf{x}}$ is known. Therefore, it is sufficient to include in the side-information only the pair $\{N_{10}, N_{11}\}$. The probability of `maybe` for each sequence, is then given by

$$\Pr\{\texttt{maybe}|T(\mathbf{x})\} = \sum_{i=0}^{\lfloor nD \rfloor} \sum_{d_0=0}^{i} \Pr\{|N_{10} - j_0| = d_0\} \cdot \Pr\{|N_{11} - j_1| = i - d_0\}. \quad (10)$$

Note that the only random variables in (10) are j_0 and j_1.

C. Representing the Side-Information

We have shown two valid (admissible) schemes with the corresponding decision rules. However, storing the side-information incurs a cost in the rate. For example, with Hamming distortion $d(\mathbf{x}, \hat{\mathbf{x}})$ can take $n+1$ different values, thus $\Delta R = \frac{1}{n}\lceil \log_2(n+1) \rceil$. For the joint type, with $\hat{\mathbf{x}}$ and n known and $\mathcal{X} = \{0, 1\}$, we have $\Delta R = \frac{2}{n}\lceil \log_2(n+1) \rceil$. Notice that this increase in the rate may not be negligible, since in general we will be interested in very low rates. For example, for $n = 1000$, ΔR is almost 0.01 bits (0.02 bits) due to the distortion (joint type). With $R = 0.1$ they represent an increased of 10% (20%). With this in mind, we consider fixed rate quantization of these quantities and modify the decision rules accordingly. Simulation results in the next section suggest the use of the distortion rather than the joint type. Thus here we focus on the scheme with distortion as side-information.

Using the k-means algorithm [8], we find 2^k decision regions representable with k bits. Note that they can be found offline according to the distribution of $d(\mathbf{x}, \hat{\mathbf{x}})$. For each distortion $d(\mathbf{x}, \hat{\mathbf{x}})$, we store only the decision region to which it belongs. Assuming $d(\mathbf{x}, \hat{\mathbf{x}}) \in [d_0 : d_1]$, the decision rule is now given by

$$g(T(\mathbf{x}), \mathbf{y}) = \begin{cases} \texttt{maybe}, & d_0 - D \le d(\hat{\mathbf{x}}, \mathbf{y}) \le d_1 + D; \\ \texttt{no}, & \text{otherwise.} \end{cases}$$

Similarly to (8), the conditional probability for `maybe` can be modified accordingly to

$$\Pr\{\texttt{maybe}|T(\mathbf{x})\} = \sum_{i=\lceil n(d_0-D) \rceil}^{\lfloor n(d_1+D) \rfloor} \binom{n}{i} 2^{-n}. \quad (11)$$

IV. Simulation results

In this section we show the performance of the proposed architectures. Specifically, we consider a dataset[1] composed of $M = 1000$ i.i.d. sequences Bern($1/2$) of length $n = 512$. As the lossy compression algorithm, we used a binary-Hamming version of the successive refinement compression scheme [9] (recall that our scheme works with *any* algorithm, therefore the actual choice of the algorithm is immaterial as long as it performs well in terms of low average distortion). For each sequence $\mathbf{x}^{(i)}$ in the database, $i \in [1 : M]$, given its signature $T(\mathbf{x}^{(i)})$ we compute the probability that $g(T(\mathbf{x}^{(i)}), \mathbf{y}) = \texttt{maybe}$, denoted by $\Pr_i(\texttt{maybe})$, analytically. We use (8) for the distortion as side-information and (10) for the joint type. In both cases we assume the query sequences are generated i.i.d. Bern($1/2$). We then compute the probability of `maybe` for the database as the average over all the sequences it contains, i.e., $\Pr\{\texttt{maybe}\} = \frac{1}{M}\sum_{i=1}^{M} \Pr\{\texttt{maybe}|T(\mathbf{x}^{(i)})\}$.

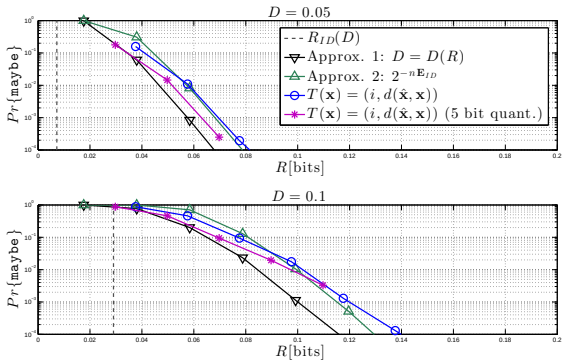

Fig. 3. Performance of the proposed architecture for $D = \{0.05, 0.1\}$ applied to a database composed of binary symmetric i.i.d. sequences.

Figure 3 shows the performance of the proposed schemes for similarity thresholds $D = \{0.05, 0.1\}$. We plot the total rate ($R + \Delta R$) vs. the probability of `maybe`. We also plot the $R_{\mathrm{ID}}(D)$ and two approximations for the probability of `maybe`. One takes the exponent approximation $2^{-n\mathbf{E}_{\mathrm{ID}}(R)}$. The other one assumes the sequences are compressed with an optimal source code, in the sense that the distortion between $\mathbf{x}$ and $\hat{\mathbf{x}}$ is constant for all the sequences, and equal to the *distortion rate function* $D(R)$ [10]. For symmetric binary sources and Hamming distortion, $D(R) = H^{-1}(1 - R)$.

As can be observed, the scheme based on the joint type does not improve the performance of the system. Therefore, hereafter we focus on the scheme with side-information given by the distortion. Based on these results, it is possible to compress the database by almost 95% ($R = 0.0575$) and retrieve on average 1% of the sequences per query, for $D = 0.05$. For $D = 0.1$, a reduction of more than 88% ($R = 0.117$) gets a probability of `maybe` of 10^{-3}, i.e., on average one sequence every 1000 is retrieved per query.

In an attempt to reduce the rate of the system while getting similar performance, we apply quantization to the distortion $d(\mathbf{x}, \hat{\mathbf{x}})$. We tested several values and found 5 bits to be the best (see Fig. 3). For example, for $D = 0.05$, we can reduce the rate by 0.01 (from 0.077 to 0.067) and still achieve probability of `maybe` close to 10^{-4}. For $D = 0.1$ and probability of `maybe` around 10^{-2}, we get a reduction of 0.008 in rate (from 0.097 to 0.089).

V. Extension to q-ary sources

The proposed scheme can be easily extended to the case of q-ary sources. Note that the decision rule of (7) still applies in this case. One important example of this kind of sources

[1]Note that the specific selection of M only represents the accuracy of the simulation results. The probability for `maybe` of the scheme, and also the fundamental limits, do not depend on M.

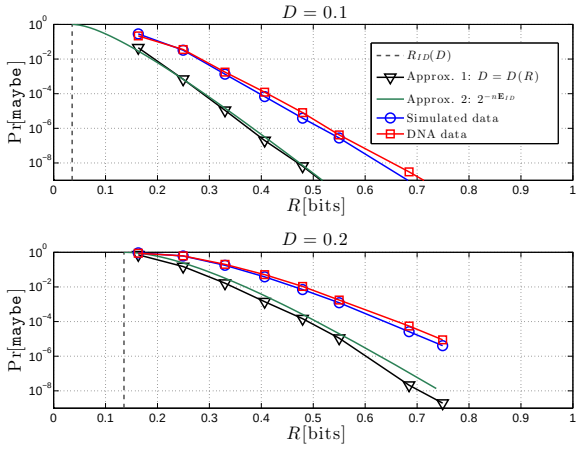

Fig. 4. Performance of the proposed architecture for $D = \{0.1, 0.2\}$ applied to two databases composed of 4-ary sequences: one generated uniformly i.i.d. and the other comprised of real DNA sequences from [2].

where this scheme would be of special importance is DNA data, where the alphabet is of size four, $\{A, C, G, T\}$.

We consider a database composed of $M = 1000$ i.i.d. uniform 4-ary sequences of length $n = 100$, and apply the proposed architecture with the lossy compression algorithm presented in [6]. To see how the scheme works on real data, we generate a database composed of 1000 DNA sequences of length 100, taken from *BIOZON* [2]. The empirical distribution is given by $p_A = 0.25, p_C = 0.23, p_G = 0.29, p_T = 0.23$. We emphasize that the proposed architecture makes the scheme *D-admissible*, independently of the probabilistic model behind the sequences of the database, if any. We consider i.i.d. and uniformly distributed query sequences to compute the probability of `maybe` in both cases, and modify (8) for 4-ary alphabet as

$$\Pr\{\texttt{maybe}|T(\mathbf{x})\} = \sum_{i=\lceil n(d(\mathbf{x},\hat{\mathbf{x}})-D)\rceil}^{\lfloor n(d(\mathbf{x},\hat{\mathbf{x}})+D)\rfloor} \binom{n}{i} 3^i 4^{-n}. \qquad (12)$$

The results for both datasets are shown in Fig. 4, together with the identification rate and the two approximations[2]. In this case, we have that $D(R)$ satisfies $R = \log_2(q) - H^{-1}(D(R)) - D(R)\log_2(q-1)$. As can be observed, the performance on the simulated data and on the DNA dataset are very similar. We present some results for the DNA database. For $D = 0.1$, we get a probability of `maybe` of 0.001 with a reduction in size of 83.5% ($R = 0.33$). For $D = 0.2$ and $R = 0.47$, we get a probability of `maybe` of 0.01.

VI. Concluding remarks

We have investigated the problem of compressing a database so that similarity queries can still be performed efficiently on the compressed database. Given a database composed of M discrete sequences $\{\mathbf{x}^{(i)}\}_{i=1}^M$ and a query sequence $\mathbf{y}$, we are interested in answering the following question: *which sequences satisfy $d(\mathbf{x}, \mathbf{y}) \leq D$?* We propose an architecture based on lossy compression algorithms that is valid for any database composed of discrete sources and queries given by distortion measures satisfying the triangle inequality. We also identify the fundamental limits for the case of a symmetric binary source and Hamming distortion under fixed-length compression. Under these assumptions, our suggested schemes exhibit on simulated data performance comparable to the fundamental limits. For example, we show that a 95% reduction in rate is possible while retrieving on average only 1% of the sequences as potentially similar (for $D = 0.05$). We also test the proposed architecture on real DNA data. We believe this architecture could be practically useful in several applications, including biological data.

[2]The identification rate exponent for q-ary sources, $q > 2$, was calculated according to [3], with an auxiliary alphabet $|\mathcal{U}| = |\mathcal{X}|$, and is therefore only an achievable result. Nevertheless, we conjecture that those are the fundamental limits, i.e., that at least for Hamming distortion, one does not need to consider $|\mathcal{U}|$ more than $|\mathcal{X}| = q$.

Acknowledgment

The authors would like to thank Golan Yona for helpful discussions. This work is supported by a grant from the Center for Science of Information (CSoI), a fellowship from the Basque government and a Google research award.

Appendix A

Proof of Theorems 1 and 2: The proof of the direct part follows directly from [3, Theorem 4]. Although the general setting of [3] is of variable length codes, in this specific case close inspection reveals that the achievability scheme used there is of fixed length. [3, Theorem 4] states that there exists a scheme that attains an exponent β for any rate $R > 1 - H(H^{-1}(\beta) - D)$. Reversing this expression results in a rate-R scheme that attains the exponent (6). It is easy to see that (6) is positive for any $R > 1 - H(0.5 - D)$. This concludes the direct part of Theorems 1 and 2.

The converse part of Theorem 2 is also a direct consequence of [3, Theorem 4], which provides a converse for the exponent. Note that a converse for any variable-length scheme is also a converse for fixed-length schemes, as in the case in our paper. In the special case of symmetric binary source and Hamming distortion, the exponents coincide. This concludes the converse part of Theorem 2.

The only case which is not a consequence of [3, Theorem 4] is the converse for Theorem 1, which we prove here. Suppose we are given an admissible scheme $(T(\cdot), g(\cdot))$ with $R < R_{\text{ID}}(D)$. The probability for `maybe` is given by

$$\Pr\{\texttt{maybe}\} = \sum_{i=1}^{2^{nR}} \Pr\{T(\mathbf{X}) = i\} \cdot \Pr\{\texttt{maybe}|T(\mathbf{X}) = i\}. \qquad (13)$$

Define A_i as the set of all sequences $\mathbf{x}$ that are mapped to i, i.e.

$$A_i \triangleq \{\mathbf{x} : T(\mathbf{x}) = i\}. \qquad (14)$$

Also, for any set $A \subseteq \{0,1\}^n$, define its *expansion* as

$$\Gamma^D(A) \triangleq \left\{\mathbf{y} \in \{0,1\}^n : \min_{\mathbf{x}\in A} d(\mathbf{x}, \mathbf{y}) \leq D\right\}. \qquad (15)$$

Since the scheme at hand is admissible, we must have

$$\Pr\{\texttt{maybe}|T(\mathbf{X})=i\} \geq \Pr\left\{\mathbf{Y}\in\Gamma^D\left(A_i\right)\right\}. \tag{16}$$

Define the Hamming ball with $\mathcal{B}_r$ as the set of all binary sequences with at most r ones:

$$\mathcal{B}_r \triangleq \left\{\mathbf{x}\in\{0,1\}^n : \sum_{i=1}^n x_i \leq r\right\}. \tag{17}$$

Note that whenever $r\geq n$ we have $\mathcal{B}_r=\{0,1\}^n$.

By the isoperimetric inequality for the hypercube (also known as Harper's theorem, see e.g. [11]), states the following. For any set B of a given number of binary sequences (viewed as points on a binary hypercube), the set that minimizes the size of the expanded set $\Gamma^D(B)$ is a Hamming ball. Formally, we have that

$$\left|\Gamma^D(A_i)\right| \geq \left|\Gamma^D(\mathcal{B}_r)\right|, \tag{18}$$

for any r that satisfies

$$|\mathcal{B}_r| < |A_i|. \tag{19}$$

Also, note that for the Hamming ball, the following facts hold:

$$|\mathcal{B}_r| = \begin{cases} \sum_{k=0}^r \binom{n}{k}, & r\leq n; \\ 2^n, & \text{otherwise}, \end{cases} \tag{20}$$

$$\Gamma^{m/n}\left(\mathcal{B}_r\right) = \begin{cases} \mathcal{B}_{r+m}, & r+m\leq n; \\ \{0,1\}^n, & \text{otherwise}, \end{cases} \tag{21}$$

$$|\mathcal{B}_r| = 2^n - \left|\mathcal{B}_{n-r-1}\right|. \tag{22}$$

Combining the above we have that for any r satisfying (19), we have

$$\begin{aligned} \Pr\{\texttt{maybe}|T(\mathbf{X})=i\} &\geq \Pr\left\{\mathbf{Y}\in\Gamma^D\left(A_i\right)\right\} && (23)\\ &= 2^{-n}\left|\Gamma^D\left(A_i\right)\right| && (24)\\ &\geq 2^{-n}\left|\Gamma^D\left(\mathcal{B}_r\right)\right| && (25)\\ &\geq 2^{-n}\left|\mathcal{B}_{r+\lfloor nD\rfloor}\right| && (26)\\ &\geq 1-2^{-n}\left|\mathcal{B}_{n-r-\lfloor nD\rfloor-1}\right|. && (27) \end{aligned}$$

Since $R<R_{\mathrm{ID}}(D)$, there exists some $\varepsilon>0$ s.t.

$$R+2\varepsilon \leq R_{\mathrm{ID}}(D)-\varepsilon. \tag{28}$$

Next, suppose that for a given i, $|A_i|>2^{n(1-R-\varepsilon)}$. In fact, this will hold for most A_i's:

$$\begin{aligned} \sum_{i:|A_i|\leq 2^{n(1-R-\varepsilon)}} \Pr\{\mathbf{X}\in A_i\} &= \sum_{i:|A_i|\leq 2^{n(1-R-\varepsilon)}} 2^{-n}|A_i| \\ &\leq \sum_{i:|A_i|\leq 2^{n(1-R-\varepsilon)}} 2^{n(-R-\varepsilon)} \leq 2^{-n\varepsilon}. \end{aligned} \tag{29}$$

The size of a Hamming ball $\mathcal{B}_r$ is given, up to polynomial factors, by $2^{nH(r/n)}$. Therefore, for large n, there exists r that satisfies (19) that satisfies

$$H(r/n) \geq 1-R-2\varepsilon. \tag{30}$$

Combining with (28), we have

$$H(r/n) \geq H(\tfrac{1}{2}-D)+\varepsilon, \tag{31}$$

or

$$r/n \geq \tfrac{1}{2}-D+\varepsilon' \tag{32}$$

for some $\varepsilon'>0$. Next, write

$$\begin{aligned} n-r-\lfloor nD\rfloor-1 &\geq n-r-nD-1 && (33)\\ &\geq \tfrac{n}{2}-\varepsilon' n-1. && (34) \end{aligned}$$

Therefore the size $\left|\mathcal{B}_{n-r-\lfloor nD\rfloor-1}\right|$ satisfies, for any $\delta>0$ and large enough n,

$$\left|\mathcal{B}_{n-r-\lfloor nD\rfloor-1}\right| \leq 2^{n(H(1/2-\varepsilon')+\delta)}. \tag{35}$$

Since δ can be chosen s.t. $H(1/2-\varepsilon')+\delta<1$, we get that

$$2^{-n}\left|\mathcal{B}_{n-r-\lfloor nD\rfloor-1}\right| \to 0, \tag{36}$$

(exponentially fast), and therefore we can write

$$\Pr\{\texttt{maybe}|T(\mathbf{X})=i\} \geq \eta_n, \tag{37}$$

where $\eta_n\to 1$ as $n\to\infty$.

Finally, combine with (29) and write:

$$\Pr\{\texttt{maybe}\} = \sum_{i=1}^{2^{nR}} \Pr\{T(\mathbf{X})=i\}\cdot\Pr\{\texttt{maybe}|T(\mathbf{X})=i\} \tag{38}$$

$$\geq \sum_{i:|A_i|>2^{n(1-R-\varepsilon)}} \Pr\{T(\mathbf{X})=i\}\cdot\Pr\{\texttt{maybe}|T(\mathbf{X})=i\} \tag{39}$$

$$\begin{aligned} &\geq \sum_{i:|A_i|>2^{n(1-R-\varepsilon)}} \Pr\{T(\mathbf{X})=i\}\cdot\eta_n && (40)\\ &= \eta_n \sum_{i:|A_i|>2^{n(1-R-\varepsilon)}} \Pr\{\mathbf{X}\in A_i\} && (41)\\ &= \eta_n\left(1-2^{-n\varepsilon}\right), && (42) \end{aligned}$$

which concludes the proof of the theorem. In fact, we have proved a *strong converse*, since the probability of `maybe` approaches 1 for $n\to\infty$. ■

REFERENCES

[1] National center for biotechnology information, the national institutes of health (NIH), genbank home page. [Online]. Available: http://www.ncbi.nlm.nih.gov/genbank

[2] A. Birkland and G. Yona, "BIOZON: a system for unification, management and analysis of heterogeneous biological data," *BMC bioinformatics*, vol. 7, no. 1, 2006.

[3] R. Ahlswede, E.-h. Yang, and Z. Zhang, "Identification via compressed data," *IEEE Trans. Inf. Theory*, vol. 43, Jan 1997.

[4] A. Ingber, T. Courtade, and T. Weissman, "Quadratic similarity queries on compressed data," in *Data Compression Conference*, 2013.

[5] ——, "Compression for quadratic similarity queries," *Submitted to the IEEE Trans. Inf. Theory*, 2013 [http://arxiv.org/abs/1307.6609].

[6] A. Gupta and S. Verdú, "Nonlinear sparse-graph codes for lossy compression," *IEEE Trans. Inf. Theory*, vol. 55, 2009.

[7] C. Gioran and I. Kontoyiannis, "Lossy compression in near-linear time via efficient random codebooks and databases," *arXiv preprint arXiv:0904.3340*, 2009.

[8] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *5th Berkeley Symp. on Math. Stat. and Prob.*, vol. 1. California, USA, 1967.

[9] R. Venkataramanan, T. Sarkar, and S. Tatikonda, "Lossy compression via sparse linear regression: Computationally efficient encoding and decoding," *CoRR*, vol. abs/1212.1707, 2012.

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & sons, 1991.

[11] B. Bollobas, *Combinatorics : set systems, hypergraphs, families of vectors, and combinatorial probability*. Cambridge Uni. Press, 1986.